

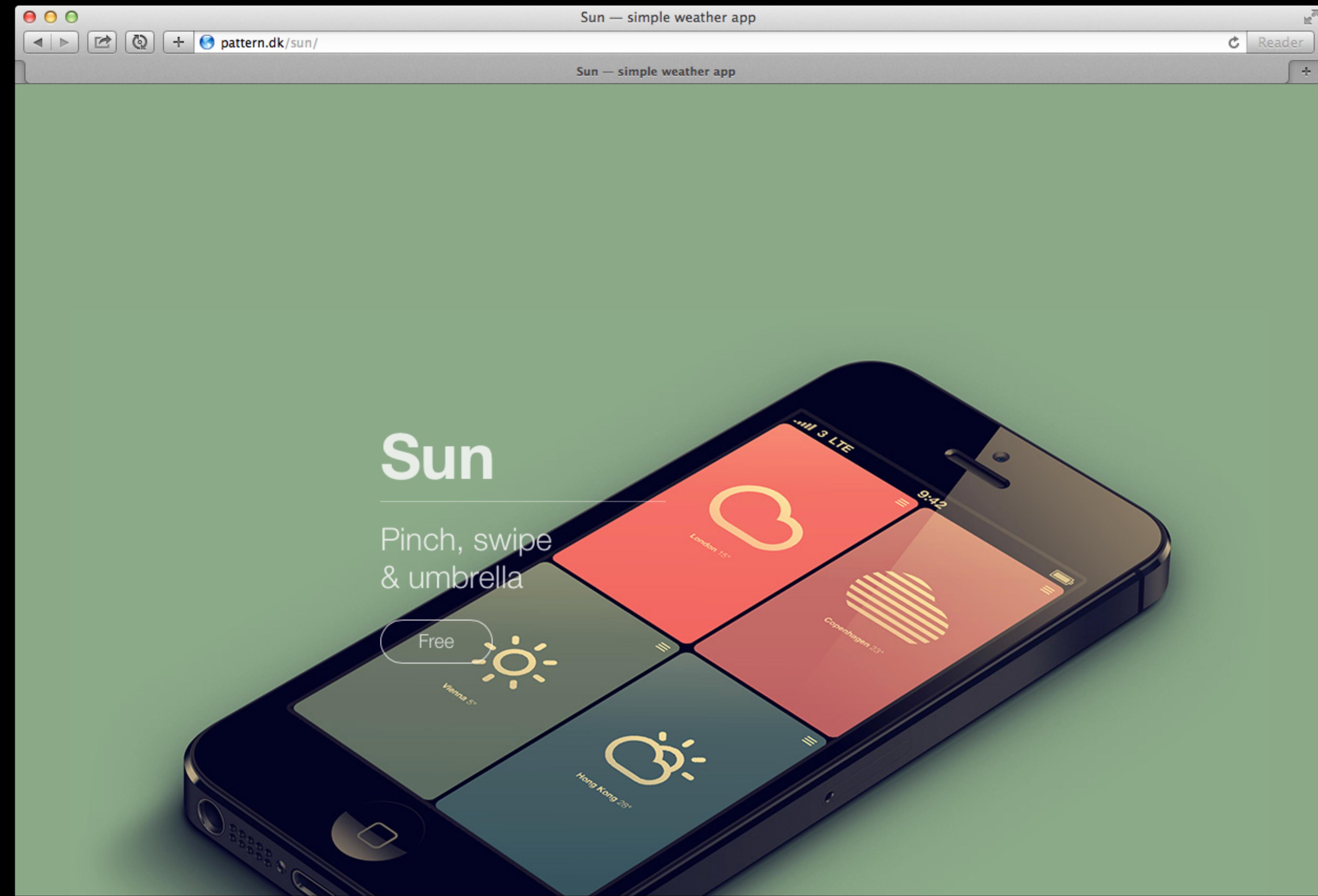
Mastering CSS  
animations and  
transitions

When was the last time something  
on the web delighted you?



<http://uxrave.com/>

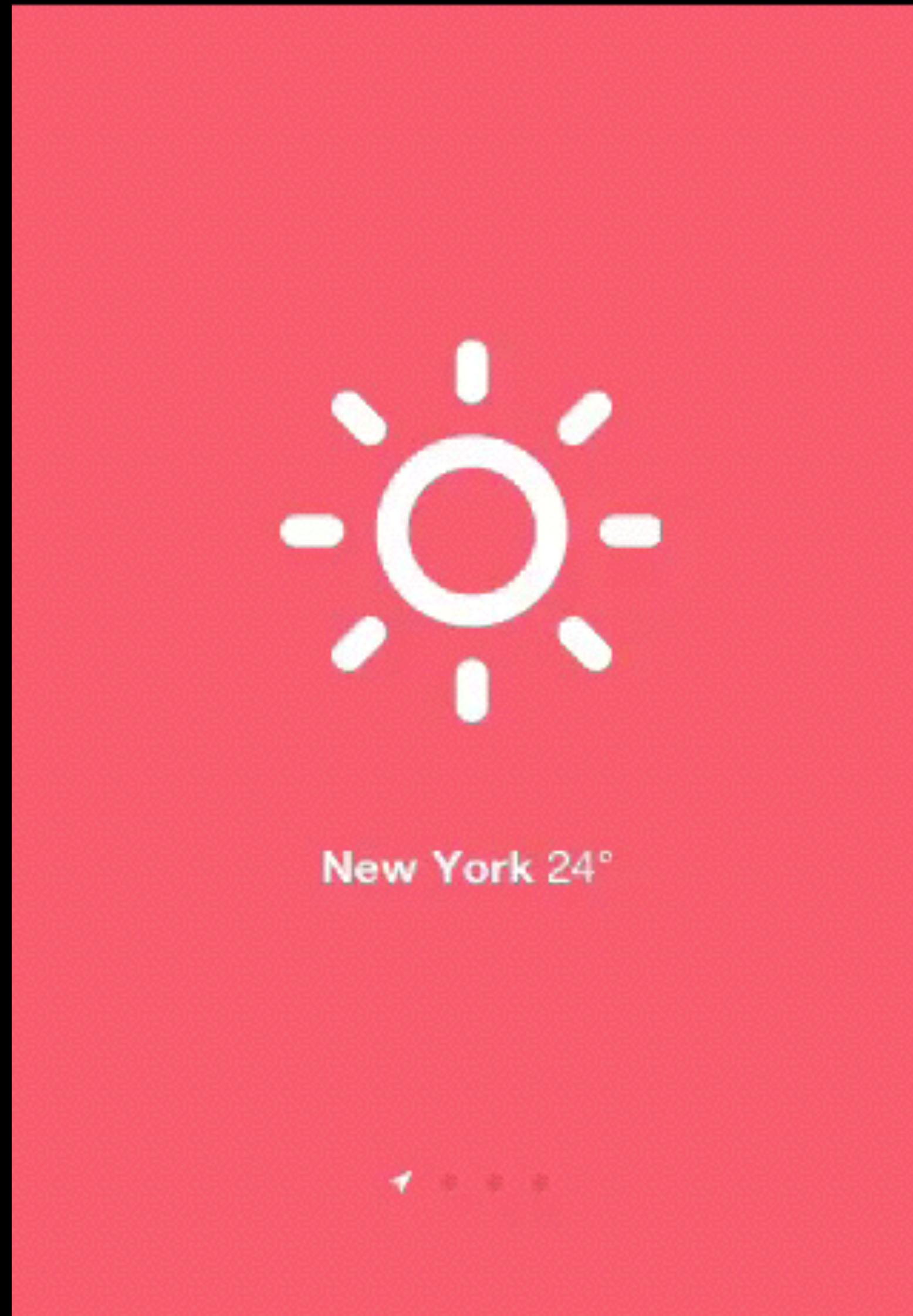
## *List Summary*



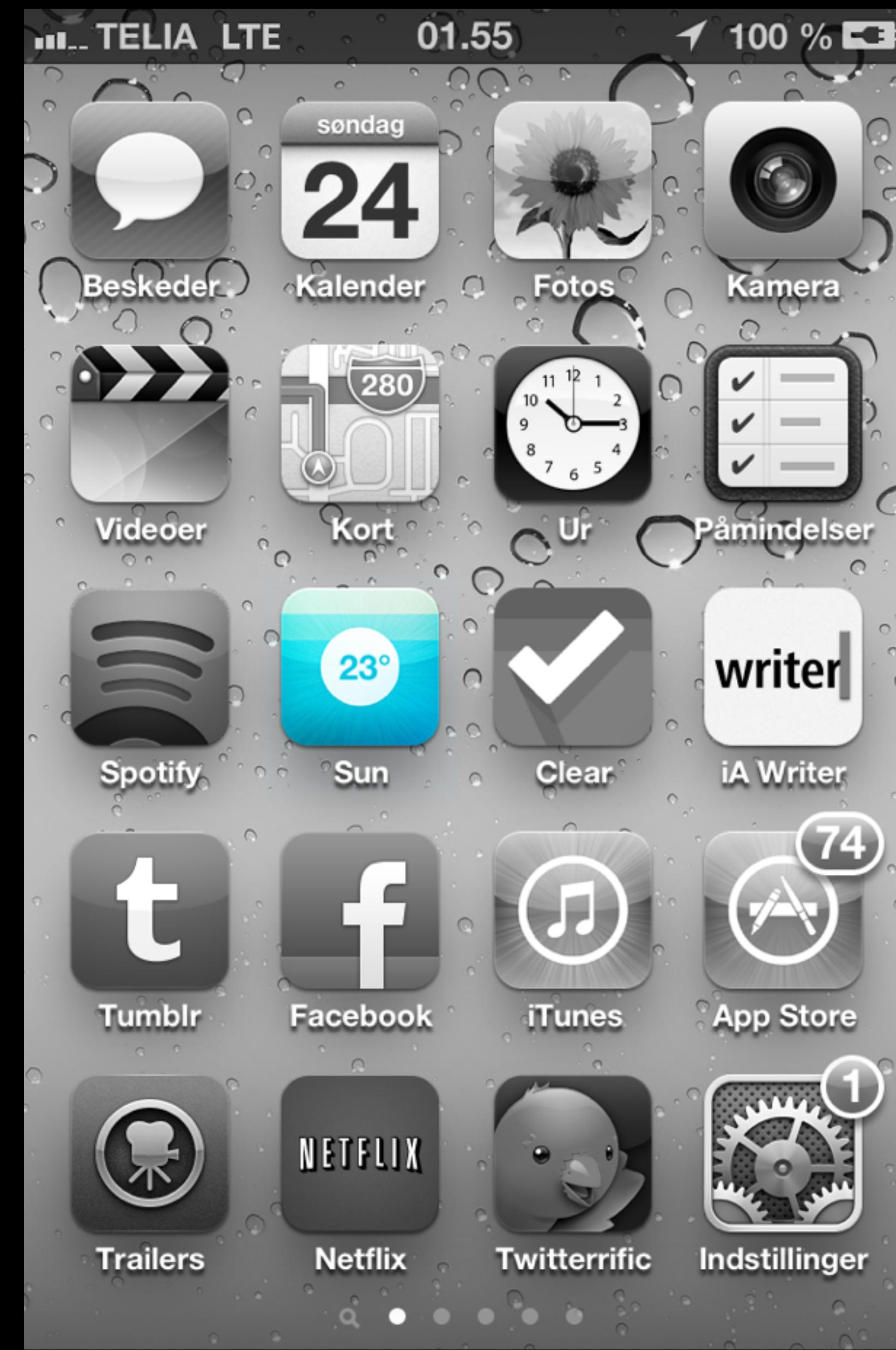
-90 T-Mobile -56

12:43 AM

22°



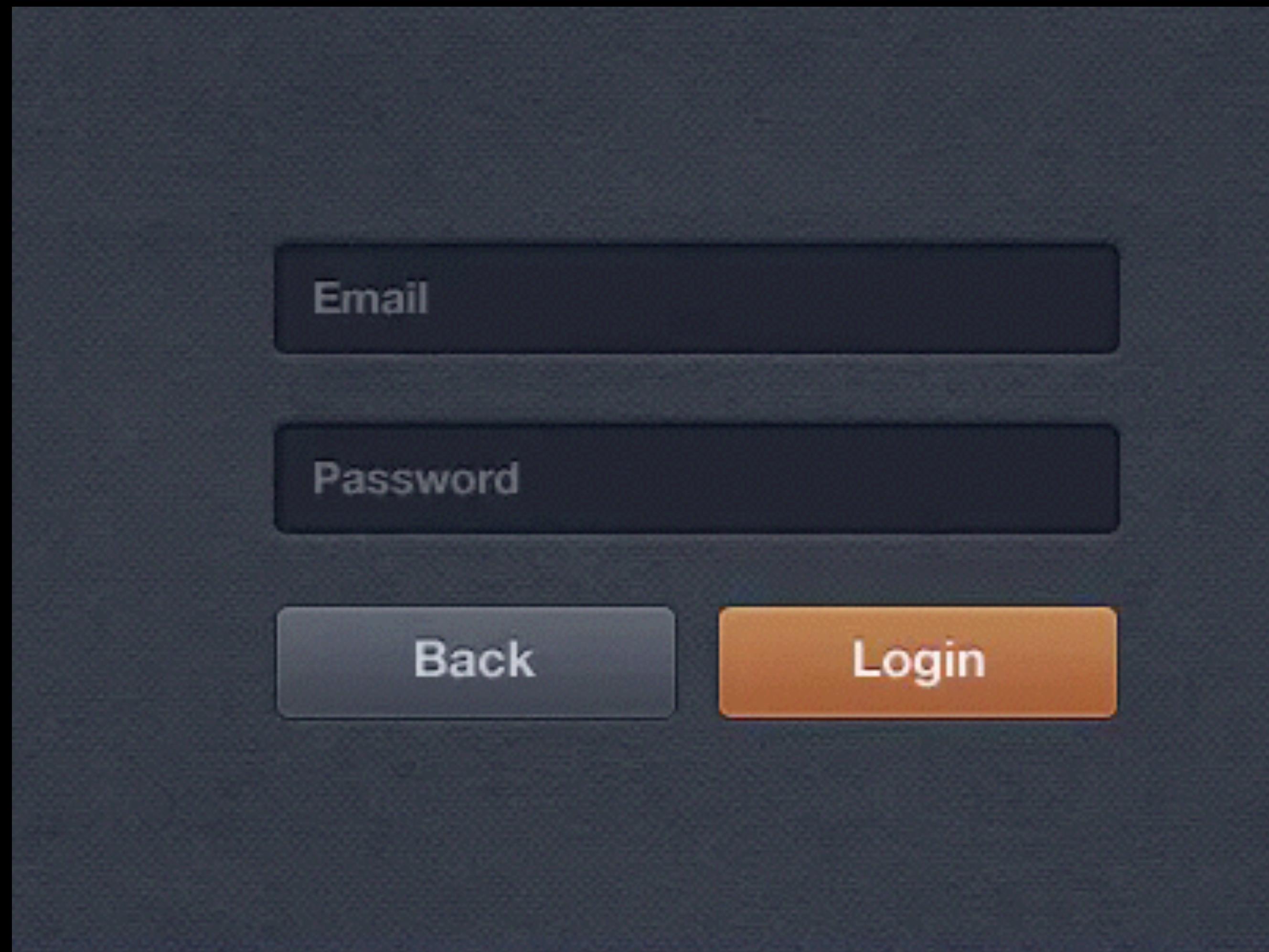
# Live icon





<http://space.angrybirds.com/launch/>

<http://joelb.me/scrollpath/>



*<http://dribbble.com/shots/419244-Login-animation>*

## **+** ADD ITEM

animations, transitions,  
transformations,  
translates

you will be creating animations  
where you call a transform that  
uses a translate function



# transitions

- an animation between changes
- properties we want to animate need to be defined explicitly
- can be triggered by adding classes or using pseudo-selectors like :hover
- example 1: change button color on hover (and animate it!)
- example 2: motion by changing the position of an element

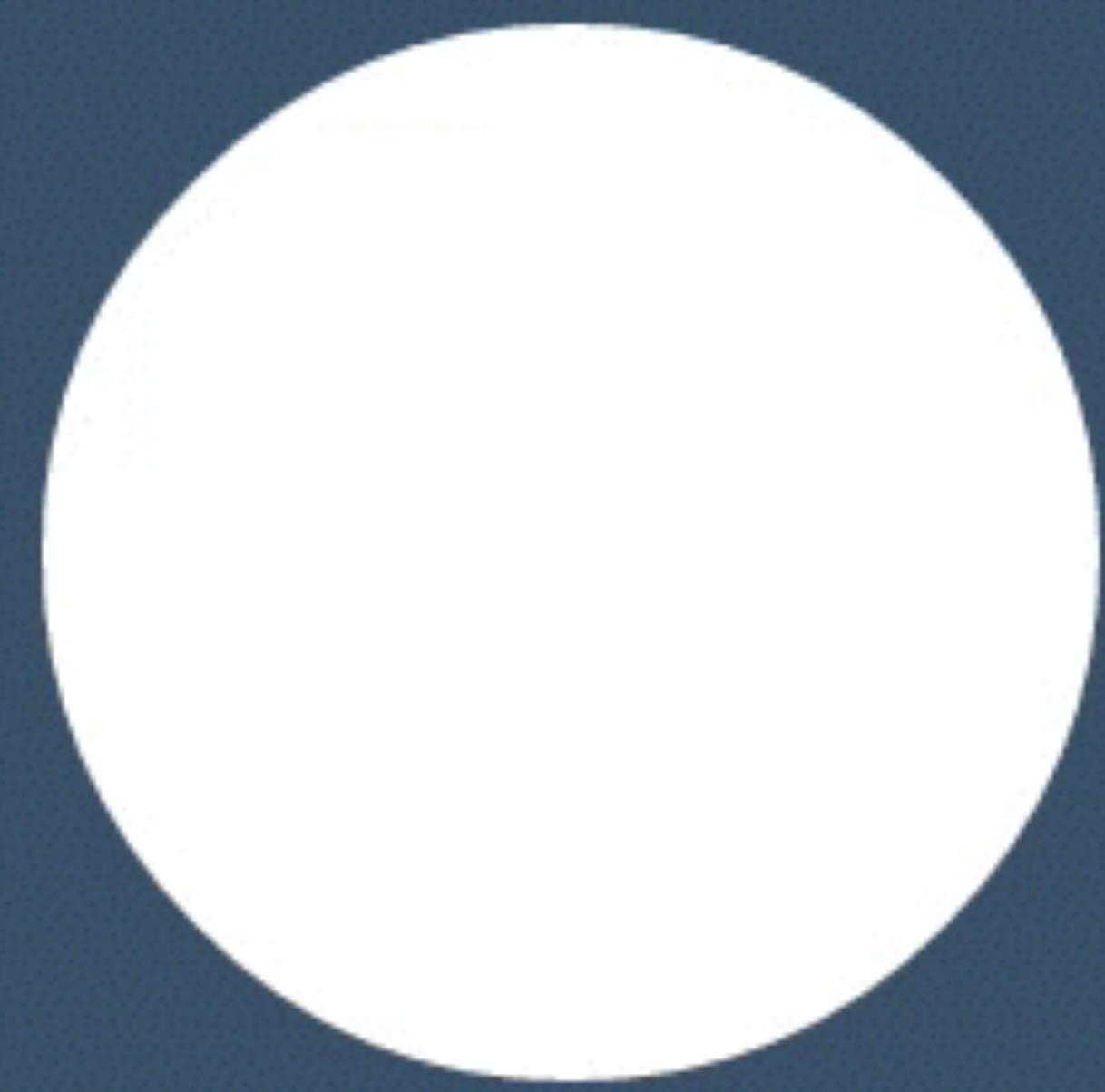
# transitions

- *transition: property duration timing-function delay;*  
*transition: background-color 5s linear 1s;*

```
<div id="circle"></div>

#circle
{
    width: 200px;
    height: 200px;
    background-color: white;
    border-radius: 50%;
    transition: height 1s linear, width 1s linear, background-color 5s linear 1s;
}

#circle:hover
{
    height: 300px;
    width: 300px;
    background-color: red;
}
```



# transitions

- transitioning properties like left and margin cause a browser to recalculate styles every frame.
- repaints (visibility) & reflows (layout)
- can be bad for performance, especially on slower mobile devices
- solution: use CSS *transformations* to enable GPU hardware acceleration where appropriate

# using translate

The screenshot shows a browser window with two tabs. The left tab displays the code for "Animate with Translate". The right tab shows the resulting animation. The animation features a laptop icon on a red background with a grid of small dots. A button labeled "add 10 more macbooks" is visible. The title "Animate with Translate" is displayed above the laptop icon.

```
<h1>Animate with Translate</h1>
<div id="animate" class=macbook></div>

h1 { padding: 90px; }
body {
  color:white;
  background:
    radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 9%, hsla(0, 100%, 20%, 0) 9% 0 0,
    radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 8%, hsla(0, 100%, 20%, 0) 10% 50px 50px,
    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 50px 0,
    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 0 50px,
    radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 50px 0.
  .text('clear')
  .click(function moar(){
    $('#[id=animate]').slice(1).remove();
  })
  .prependTo('body')
}

var machines = 10;
```

```
1 <h1>Animate with Top/Left</h1>
2 <div id="animate" class=macbook></div>
3
4 h1 { padding: 90px; }
5 body {
6   color:white;
7   background:
8     radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 9%, hsla(0, 100%, 20%, 0) 9% 0 0,
9     radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 8%, hsla(0, 100%, 20%, 0) 10% 50px 50px,
10    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 50px 0,
11    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 0 50px,
12    radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 50px 0.
13
14 var machines = 10;
```

# using top/left coordinates

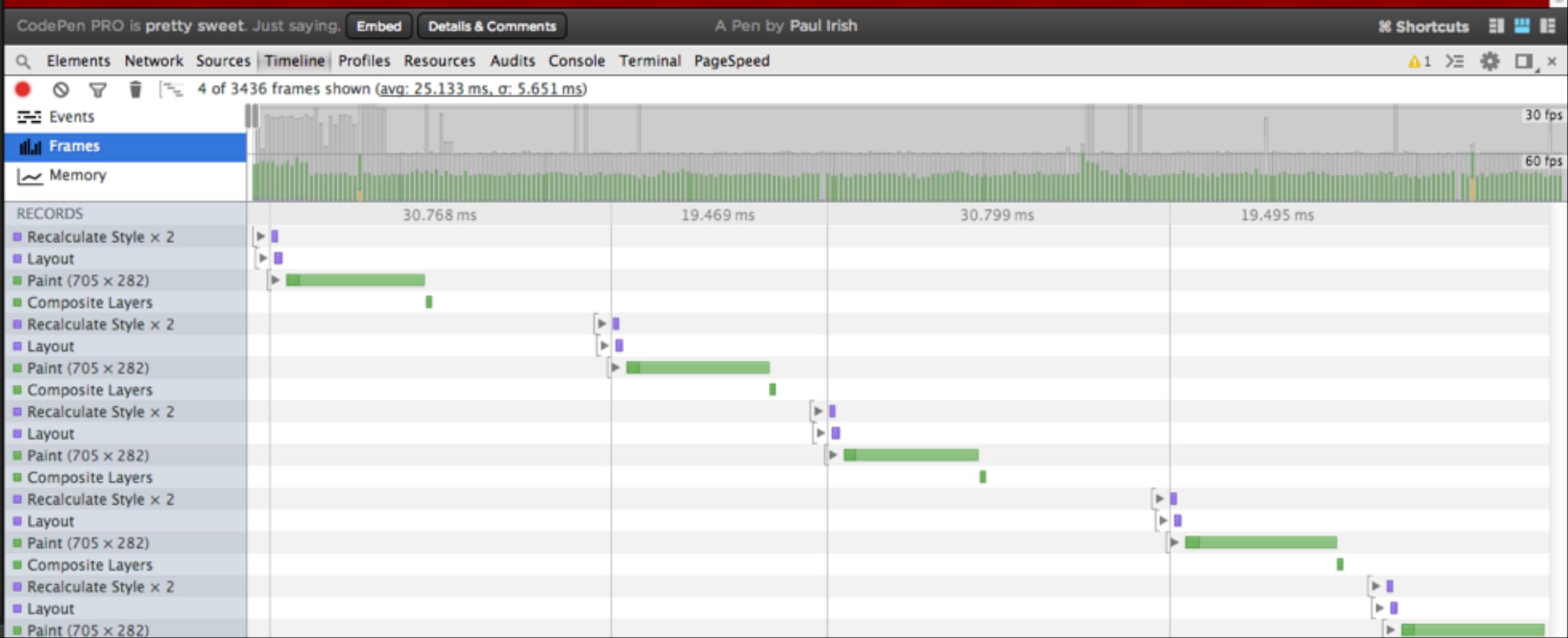
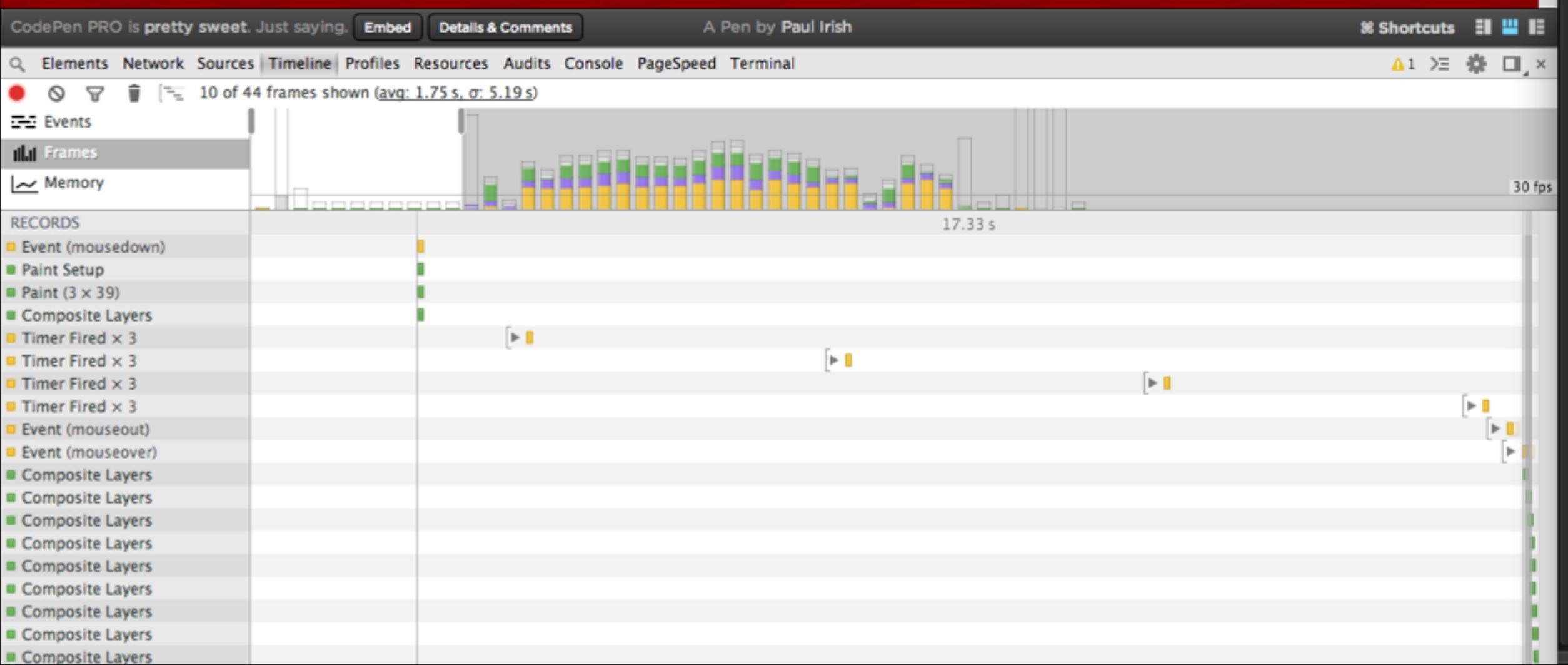
The screenshot shows a browser window with two tabs. The left tab displays the code for "Animate with Top/Left". The right tab shows the resulting animation. The animation features a laptop icon on a red background with a grid of small dots. A button labeled "add 10 more macbooks" is visible. The title "Animate with Top/Left" is displayed above the laptop icon.

```
<h1>Animate with Top/Left</h1>
<div id="animate" class=macbook></div>
<button>clear</button>
<button>add 10 more macbooks</button>

h1 { padding: 90px; }
body {
  color:white;
  background:
    radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 9%, hsla(0, 100%, 20%, 0) 9% 0 0,
    radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 8%, hsla(0, 100%, 20%, 0) 10% 50px 50px,
    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 50px 0,
    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 0 50px,
    radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 50px 0.
  .text('clear')
  .click(function moar(){
    $('#[id=animate]').slice(1).remove();
  })
  .prependTo('body')
}

var machines = 10;
```

```
1 <h1>Animate with Translate</h1>
2 <div id="animate" class=macbook></div>
3
4 h1 { padding: 90px; }
5 body {
6   color:white;
7   background:
8     radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 9%, hsla(0, 100%, 20%, 0) 9% 0 0,
9     radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 8%, hsla(0, 100%, 20%, 0) 10% 50px 50px,
10    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 50px 0,
11    radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 0 50px,
12    radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 50px 0.
13
14 var machines = 10;
```



# transform(ation)s

- physically change the look of an element
- example 1: change the size of an element
- example 2: rotate/spin an element
- example 3: move an element
- can be 2D or 3D

# transitions with transforms

```
<div id="shape"></div>
```

```
#shape
{
    width: 150px;
    height: 150px;
    background-color: #e74c3c;
    transition: -webkit-transform 1s;
}
```

```
#shape:hover
{
    -webkit-transform: scale(2) translateY(100px) rotate(45deg);
}
```



you will be creating animations  
where you call a transform that  
uses a translate function



# animations vs. transitions

transitions	animations
go from A to B	got from A (over B, C, D, E) to ... via keyframes
need to be triggered somehow	can start automatically and loop
you have to be formal about which properties you want to animate	

Transforms are ways to change your shapes with scale(), rotate(), skew(),...

# 2D transforms

- translate()
- rotate()
- scale()
- skew()
- matrix() = all of the above

```
<div id="square"></div>
```

```
#square
{
    width: 200px;
    height: 200px;
    background-color: white;
    border-radius: 10%;
    margin-left: 0px;
    transition: -webkit-transform 0.5s;
}
```

```
#square:hover, #square.automatic
{
    -webkit-transform: scale(2.2) translate(200px, 100px) skew(10deg)
rotate(45deg);
}
```



# animations

- keyframes
- can loop
- can start automatically

```
<div id="square" class="automatic"></div>
```

```
.automatic
```

```
{  
  -webkit-animation: moveIt 2s infinite;  
}
```

```
@-webkit-keyframes moveIt
```

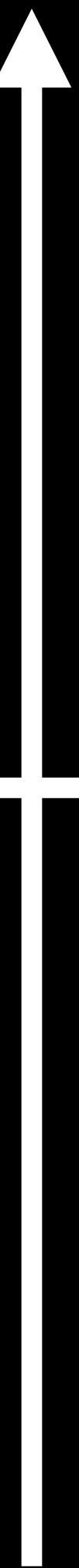
```
{  
  0%    {}  
  25%   { -webkit-transform: rotate(45deg); }  
  50%   { left:50%; top: 50%; -webkit-transform: scale(2.5) rotate(45deg); }  
  100%  { -webkit-transform: rotate(0deg); }  
}
```



# 3D transforms

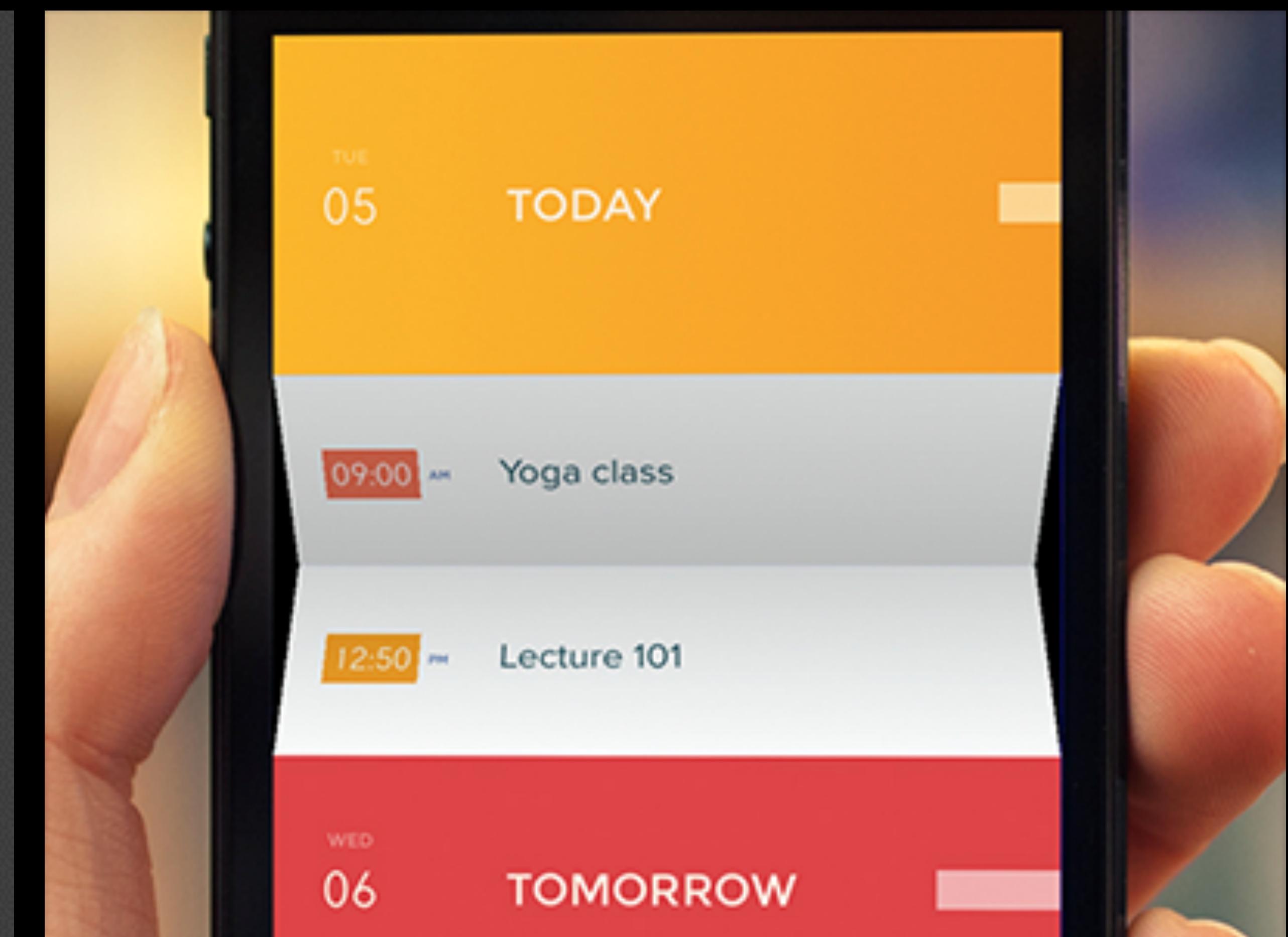
- -webkit-transform: perspective(300px) rotateY(20deg);
- perspective() sets the angle of the view
  - the higher the value the further you are away from the element
  - the lower the value the closer you are to the element
- you will be using rotateY() and rotateX() the most
- x,y,z

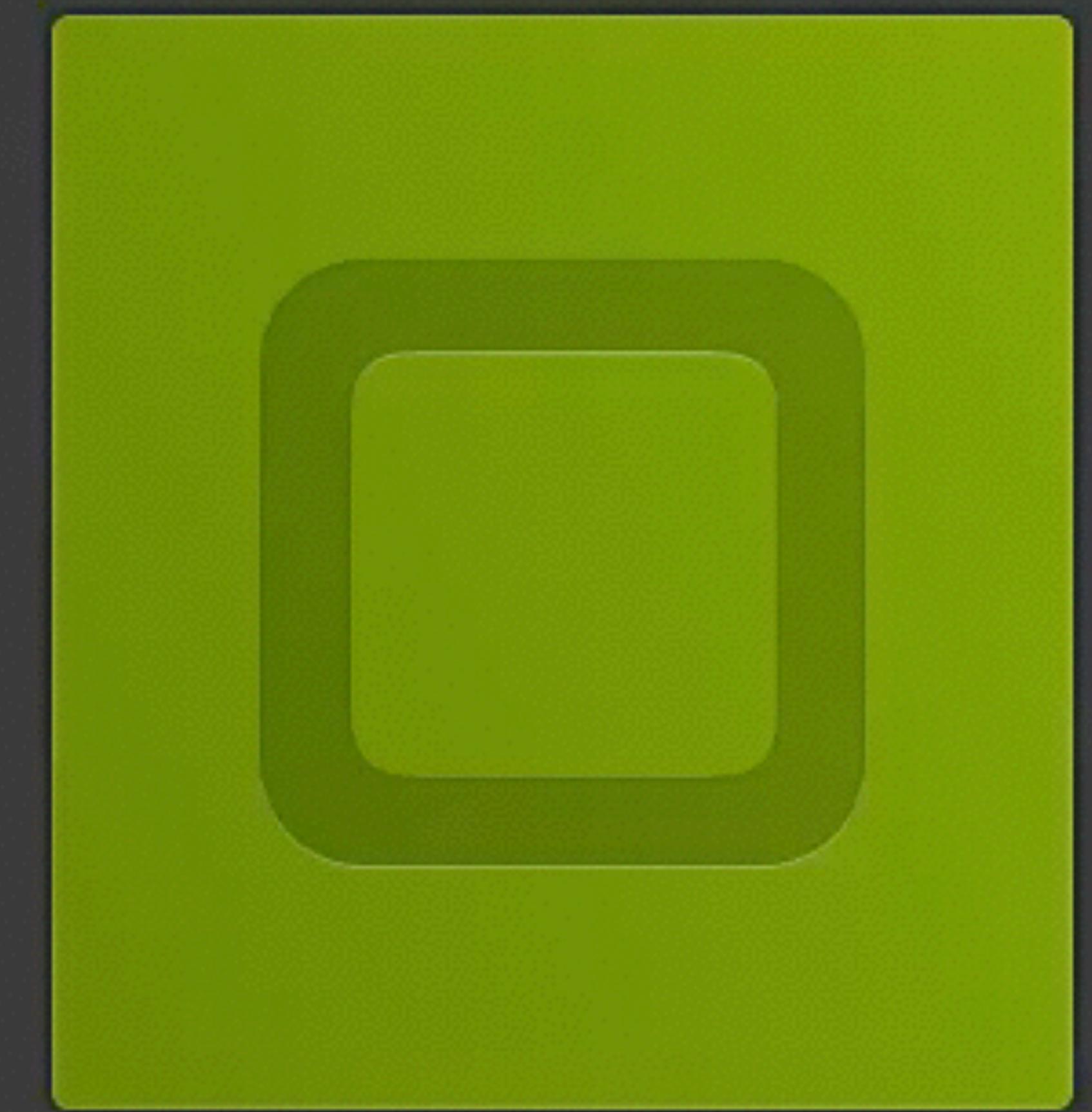
y-axis



x-axis







<http://flisterz.com/foldcover/>

FOLD/UNFOLD

Element 1

Element 2

Element 3

Element 4

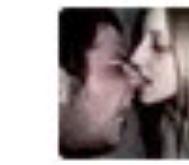
Element 5

# Buddycloud



**Vera** from Campus Library 3 days  
Pretend. You pretend the feelings are there, for the world, for the people around you.

[See 5 More Posts](#)



**Gero** from Regensburg 3 days  
I feel like a jigsaw puzzle missing a piece. And I'm not even sure what the picture should be.



**Betty** from Deggendorf 3 days  
I'm going to tell you something that I've never told anyone before.

post a comment...



**Vera** from Home 5 days  
Night time - sympathize - I've been working on white lies. So I'll tell the truth - I'll give it up to you. And when the day come, it will have all been fun. We'll talk about it soon.

post a comment...

<https://developer.mozilla.org/en-US/demos/detail/paperfold-css/launch>

```
<div id="element" class="flipInY"></div>
```

```
.flipInY
```

```
{
```

```
  -webkit-transform-origin: left bottom;  
  -webkit-animation: flipInY 1s;  
  -webkit-animation-fill-mode: forwards;
```

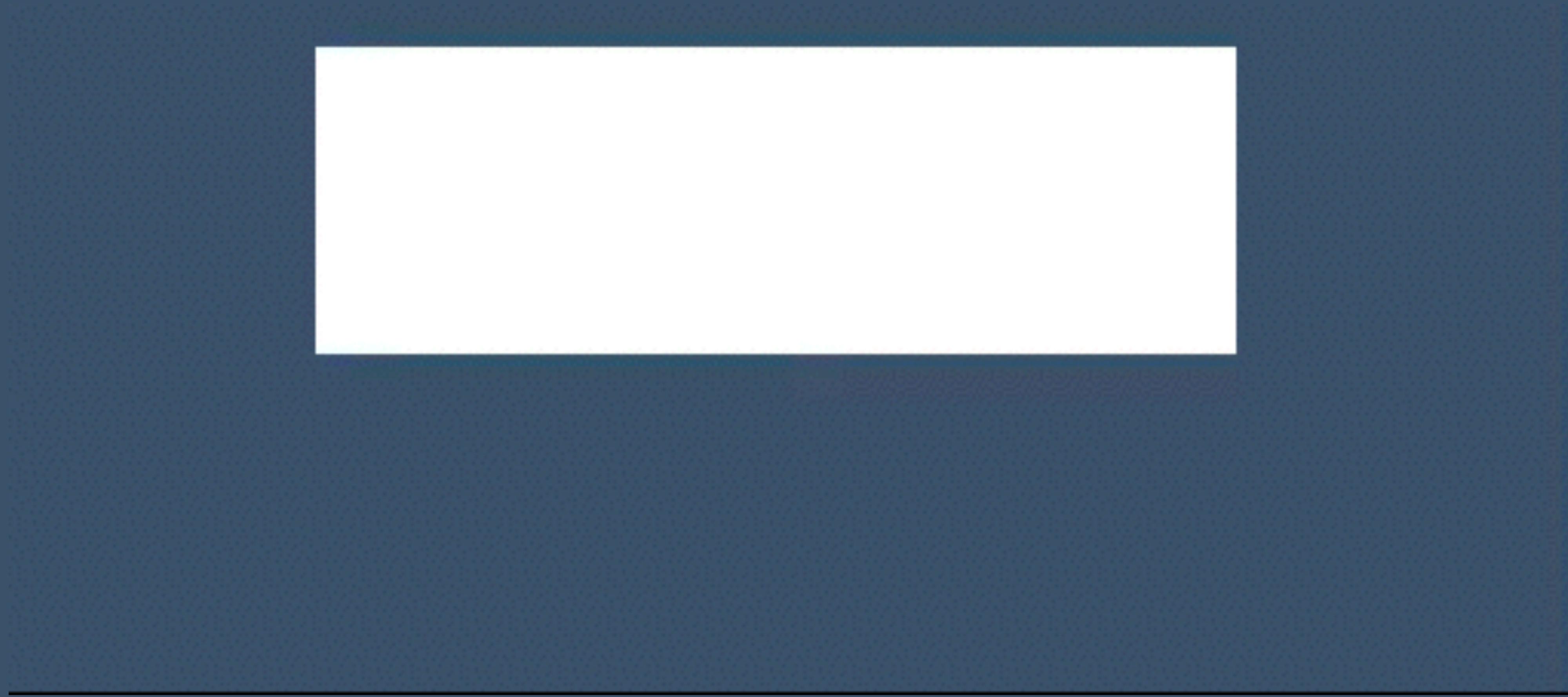
```
}
```

```
@-webkit-keyframes flipInY
```

```
{
```

```
  0%   { -webkit-transform: perspective(400px) rotateY( 90deg ); opacity: 0; }  
  40%  { -webkit-transform: perspective(400px) rotateY( -10deg ); }  
  70%  { -webkit-transform: perspective(400px) rotateY( 10deg ); }  
  100% { -webkit-transform: perspective(400px) rotateY( 0deg ); opacity: 1; }
```

```
}
```



---

# Can I use CSS3 3D Transforms?

[View in interactive mode](#)

*Compatibility table for support of CSS3 3D Transforms in desktop and mobile browsers.*

= Supported   = Not supported   = Partially supported   = Support unknown

CSS3 3D Transforms - Working Draft

### *Method of transforming an element in the third dimension*

Resources: [Multi-browser demo](#) [WebPlatform Docs](#) [Mozilla hacks article](#) [3D CSS Tester](#) [has.js test](#)

### **Global user stats\*:**

Support:	66.04%
Partial support:	10.46%
Total:	76.5%

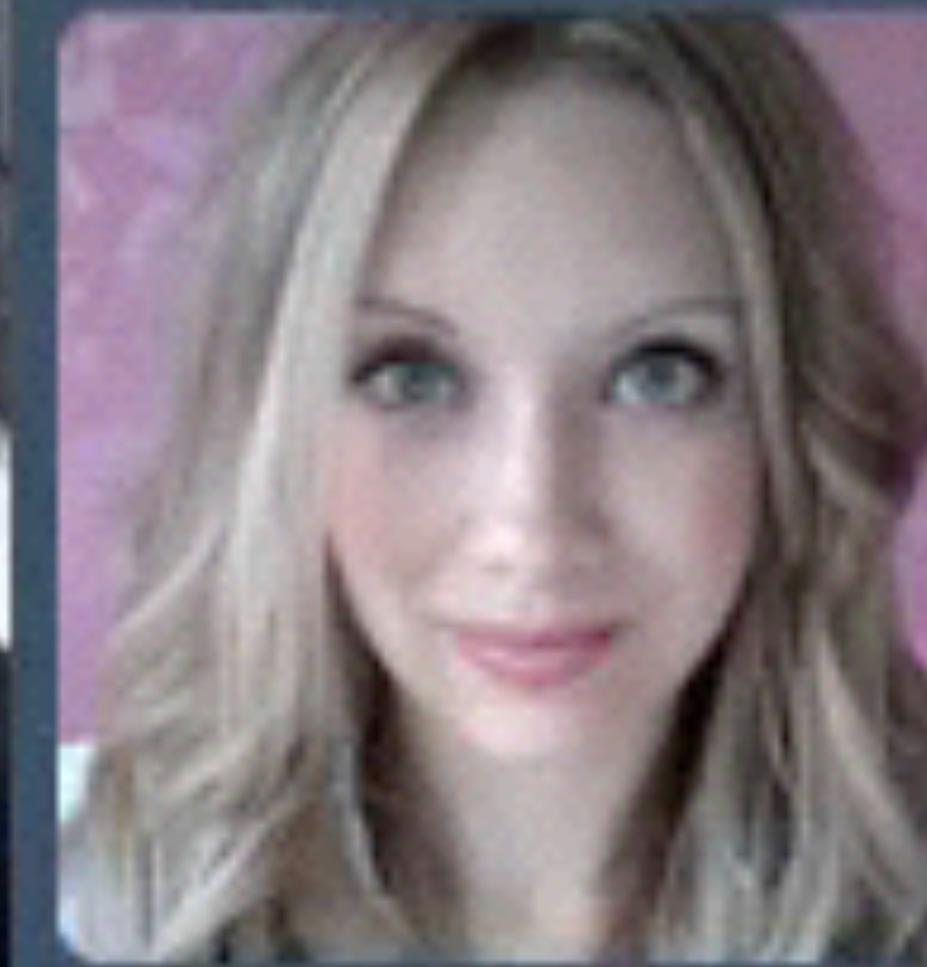
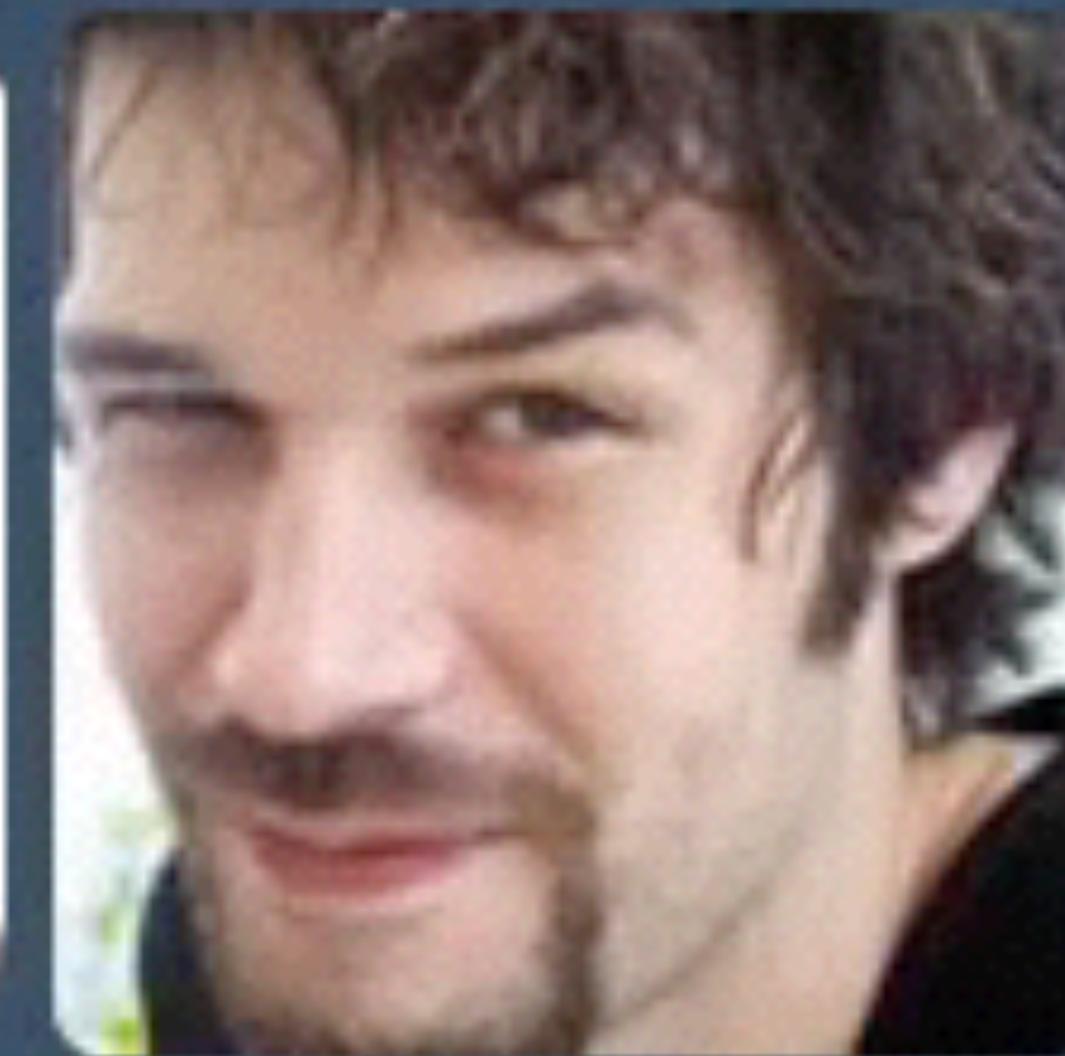
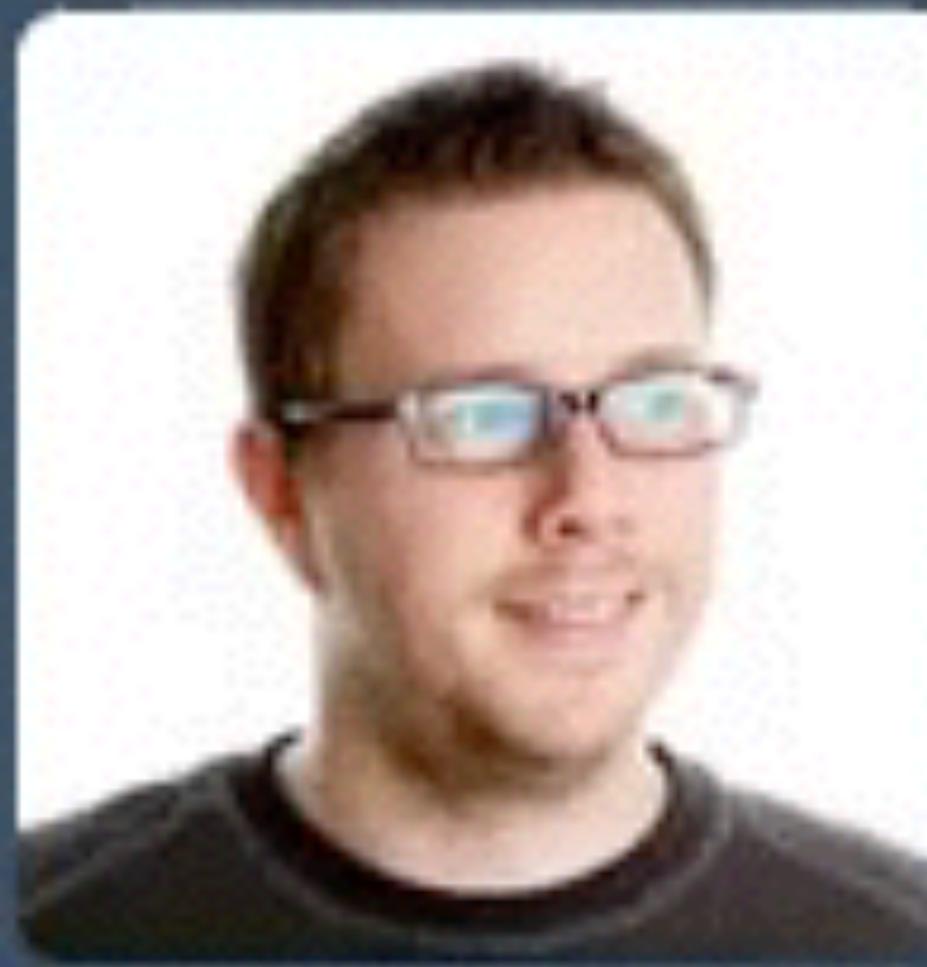
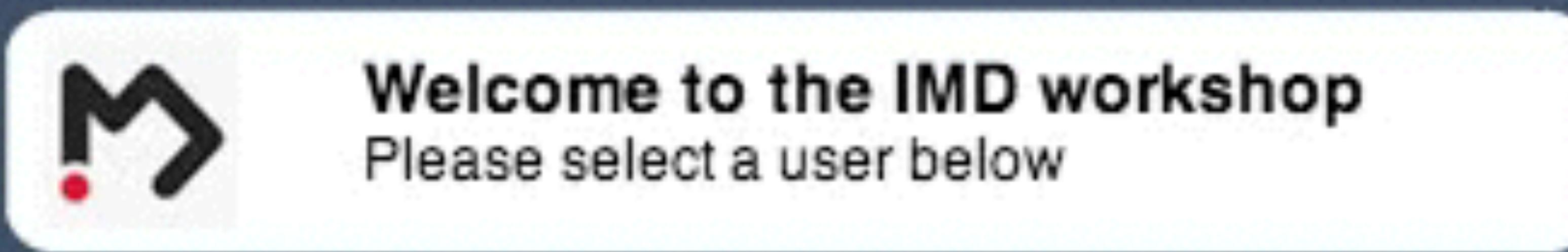
you will be creating animations  
where you call a transform that  
uses a translate function



# Get to work: animation1



# Get to work: animation2



pitfalls

# web dev hell



**The harder I try, the worse I get**

# browser support

- do I need vendor prefixes?
- fallbacks? jQuery's `.animate()`?
- transforms on older browsers: cssSandpaper
- performance (especially on mobile)
  - test test test!
  - use transforms instead of top/left/margin

Modernizr: the feature detection library for HTML5/CSS3

modernizr.com

Reader

# Modernizr

FRONT-END DEVELOPMENT DONE RIGHT

DOWNLOAD DOCUMENTATION RESOURCES NEWS

“An indispensable tool.”  
— Bruce Bowman, sr. product manager, Edge Tools & Services

**Modernizr** is a JavaScript library that detects HTML5 and CSS3 features in the user's browser.

## Why use Modernizr?

Taking advantage of cool new web technologies is great fun, until you have to support browsers that lag behind. Modernizr makes it easy for you to write conditional JavaScript and CSS to handle each situation, whether a browser supports a feature or not. It's perfect for doing progressive enhancement easily.

## How it works

Modernizr runs quickly on page load to detect features; it then creates a JavaScript object with the results, and adds classes to the `html` element for you to key your CSS on. Modernizr supports dozens of tests, and optionally includes [YepNope.js](#) for conditional loading of external `.js` and `.css` resources.

Check out the [full list of features](#) that Modernizr detects, or learn more about [conditional resource loading with Modernizr](#).

### Download Modernizr 2.7.1

Use the commented, uncompressed Development version to develop with and learn from.

Then, dive into the Production build tool and pick just the tests you need!

[View documentation](#)

**DEVELOPMENT**  
Uncompressed, 42 Kb

**PRODUCTION**  
Configure Your Build

-  Follow us on Twitter
-  Contribute on GitHub
-  Subscribe with RSS
-  Donate to Modernizr

## Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- [Documentation: Getting started](#)
- [Taking Advantage of HTML5 and CSS3 with Modernizr, Faruk Ateş](#)
- [How to use Modernizr, Inayaili de León](#)
- [Modernizr: front-end development done right, Ryan Seddon](#)
- [wiki] [The Undetectables: features that cannot be detected](#)
- [wiki] [Cross-browser Polyfills](#)

Also check out our [Resources section](#).

**Tip:** check our [Modernizr test suite](#) to quickly test your current browser's features.

## Latest news

- April 2nd, 2013  
[Modernizr 3, Stickers & Diversity](#)  
News on v.3, stickers, and a message from the team.

### PREVIOUS

- January 7th, 2013  
[Modernizr Policy on Browser Bugs and Feature Detects](#)  
September 4th, 2012  
[Modernizr 2.6.2 released](#)  
July 19th, 2012  
[Modernizr 2.6 released](#)

# the 300ms delay syndrome

1. touchstart
2. touchend
3. wait 300ms to know if there is a double tap or not
4. click

<http://www.youtube.com/watch?v=AjUpiwvla5A>

# fastclick.js

- <https://github.com/ftlabs/fastclick> = AWESOME
- removes the 300ms delay
- some browsers are already adapting their default behaviour when detecting viewport tag

# considerations

- does the animation distract from the information it should be supporting?
- does your solution work on *\*any\** device?
- what's the worst case scenario?

# must read

- <http://www.sitepoint.com/advanced-css3-2d-and-3d-transform-techniques/>
- <http://learn.shayhowe.com/advanced-html-css/css-transforms>
- <http://learn.shayhowe.com/advanced-html-css/transitions-animations>
- <http://css3 bradshawenterprises.com/transitions/>
- [http://www.kirupa.com/html5/css3\\_animations\\_vs\\_transitions.htm](http://www.kirupa.com/html5/css3_animations_vs_transitions.htm)
- Bonus inspiration: <http://uxrave.com/>

# to pass this level you

- have read the articles from the “must read” section
- you have successfully built the two animations described in the slides „get to work”
- you have pushed your code to your GitHub account where you will place all of your work for this course
- this assignment can also be found on toledo